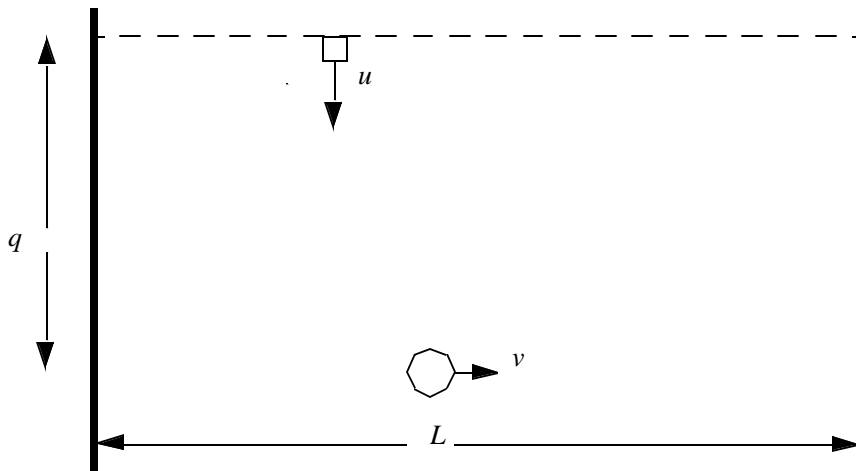


Project 2: Discrete Event Simulation of a Barrier Search

Problem Description

Targets attempt to travel down a channel having width L while a searcher creates a “barrier” by traveling back and forth perpendicular to the channel. Each target has speed u and the searcher has speed v . The searcher is equipped with a sensor that has range R and continuous detection time that is exponential with a mean of λ . The targets start at a location that is uniformly distributed along the width of the channel and distance q from the searcher’s track. A new target starts every t time units, where t is gamma distributed with parameters α and β .



The objective is to estimate the average proportion of targets detected over a period of 30 days (assume that the searcher can continuously patrol 24 hours a day). Perform 50 independent replications to produce an interval estimate.

Parameters

Use the following values for the model.

- Channel width (L) = 300 nm
- Searcher speed (v) = 30 knots
- Sensor Range (R) = 15 nm
- Mean detection time when in range (λ) = 3 min
- Target speed (u) = 25 knots
- Initial distance from searcher (q) = 100 nm
- Interarrival time parameters for targets (hours): $\alpha = 2.0$, $\beta = 1.5$

Deliverables

1. Your well-commented source code
2. Output (non-verbose)
3. Brief (1-2 pages) write-up describing the approach you took and the design of your model.

Hints

1. Define a class called `TargetGenerator` that instantiates a target, registers it with a referee, and listens for the `EndMove` of that target, as with the `Mover Manager` classes. When it hears the `EndMove` event, unregister the target from the referee and update a counter.
2. Define a class that listens for either the `detection PropertyChangeEvent` or the `Detection` event of the `Sensor` and updates a counter.
3. Use the `StudentT.getQuantile(...)`, in the `simkit.stat` package to obtain the quantile for your interval estimate.